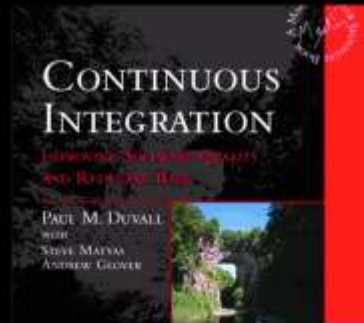


Continuous Integration

.NET Integration with Hudson and NAnt

Todd H. Gardner
The Toro Company
Todd.gardner@toro.com



Survey for:

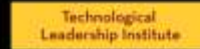
Familiar with Continuous Integration?
Familiar with NAnt/Ant Scripting?
Familiar with Hudson?

About Me

Bachelor of Science, Computer Science
Minnesota State University Mankato



Master of Science, Management of Technology
University of Minnesota



Senior Software Developer
The Toro Company



Count on it.

4/11/2010

Agenda | Continuous Integration

Concepts of Continuous Integration

Build Server Scripting

Build Integration

Build Metrics

Build Deployment

4/11/2010

Introduce the agenda

Agenda | Continuous Integration

Concepts of Continuous Integration

4/11/2010

First, we'll cover some concepts of continuous integration, and what it can do for you.

Continuous Integration

A software development practice where teams integrate work frequently, multiple integrations per day. Each is verified by an automated build to detect integration errors quickly.

--Martin Fowler

4/11/2010

Benefits and Barriers

Benefits

- Reduce Risk
- Reduce Repetition
- Deployable Software
- Better Visibility
- Greater confidence

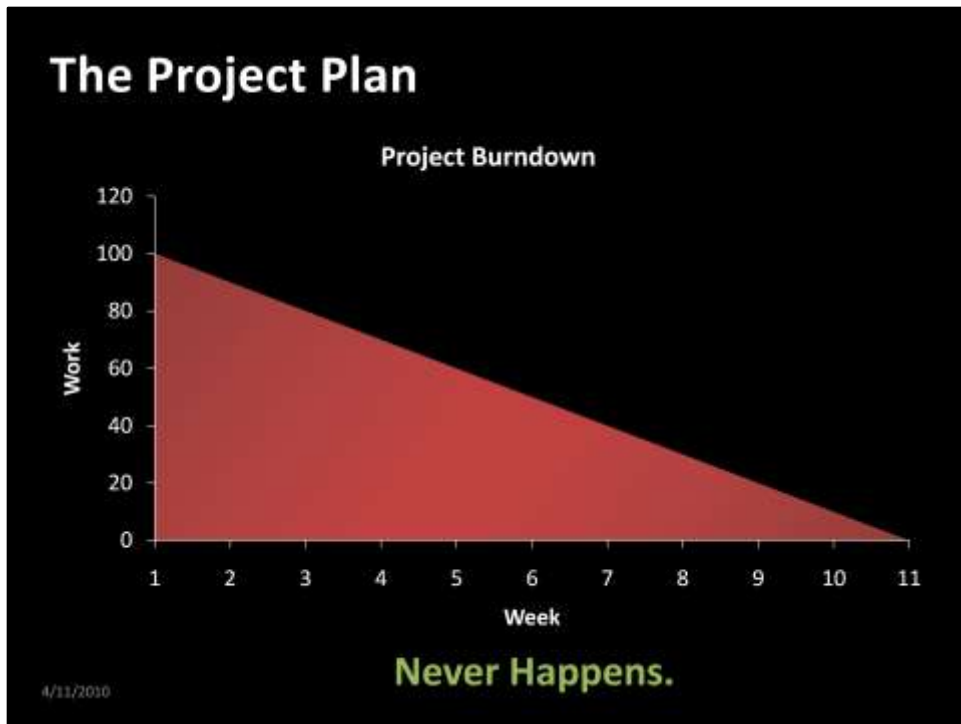
Barriers

- Increased Overhead
- Big Behavior Change
- Investment Costs
- Investment Time

4/11/2010

Some of what I will show today will help reduce your investment costs and investment time.

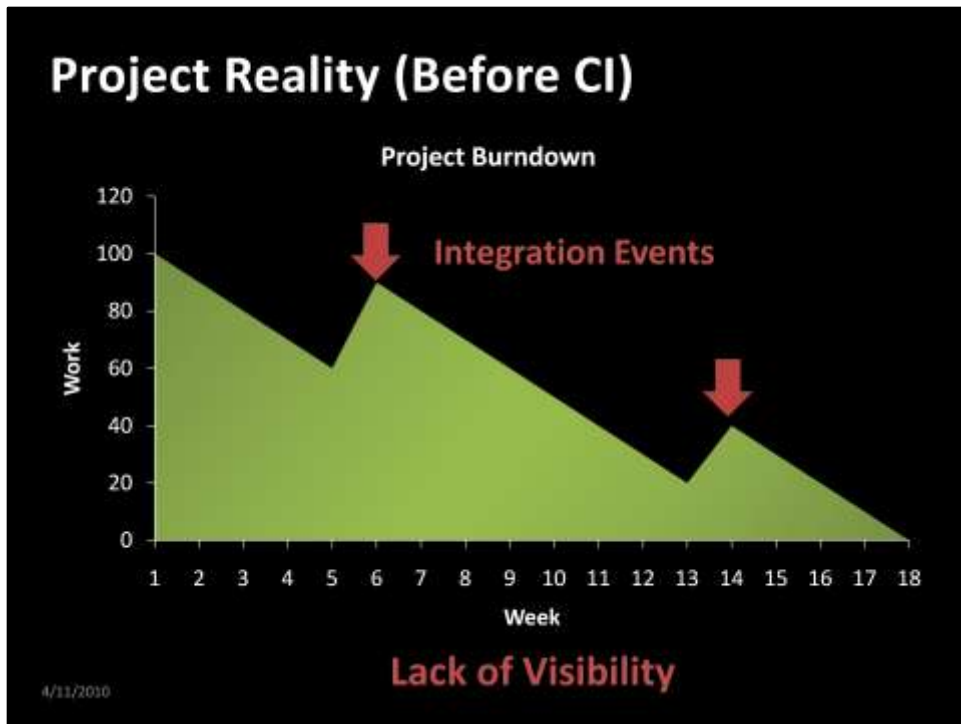
OnFinish – now, I'll show you my favorite scenario for Continuous Integration.



Project Burndown chart – this is the plan that we develop at the beginning of a project

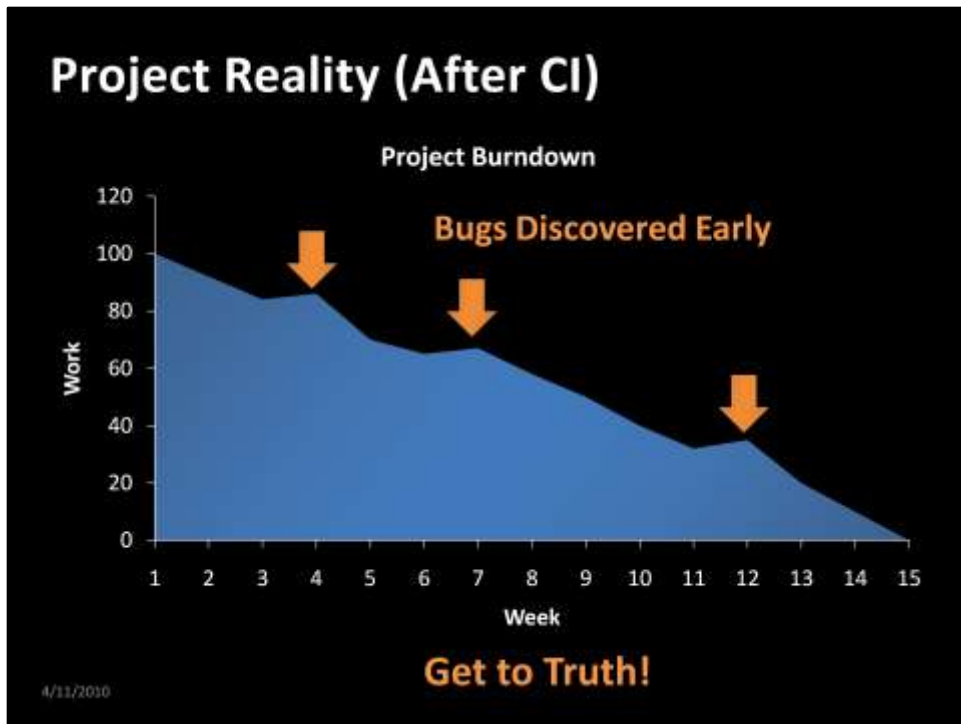
- How much work we have
- What our capacity is
- When we think we'll be done

- But it Never happens!



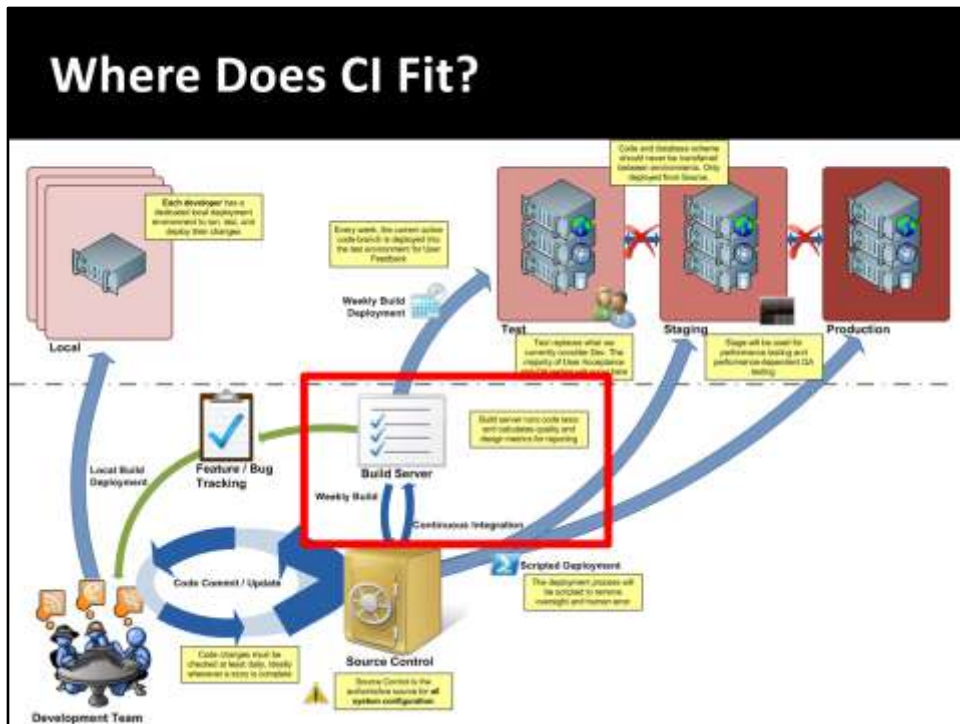
Before CI, this is how it went.

- When we 'get around' to integrating code, found lots of new work (missing functions, bugs, etc.)
- I have no idea how much work is left



Continuous Integration Exposes issues quickly!

- Bugs and defects discovered daily
- Clear visibility of what is left to do



Picture I use to talk about my development cycle, I use Hudson as my build server to do

- Continuous Integration
- Weekly Builds
- Weekly Deployments

Hudson pulls from **Source Control**, builds **locally**, then deploys into my **test environment**

Why Hudson?

- **Many** Other Tools:
 - Cruise Control.NET
 - Team Foundation Server
- Hudson
 - Free and Open Source
 - Easy to Setup and Install
 - Vibrant plug-in community
 - Management-friendly reports
- Easy to Change Later

4/11/2010

Initially chose Hudson because we are a mix of Java and .NET – worked well for both
But we are a small team, and can't afford the overhead of big tools like TFS

Hudson was easy and extensible

With how our projects are constructed, Hudson is just a batch runner, and report
shower – I could change to CC.NET tomorrow if I needed to

Demo Environment

- Windows Server 2008 SP2
 - IIS 7 Role
 - SQL Server 2008 Developer Ed.
- Visual SVN Repository
 - Tortoise SVN Client
- Visual Studio 2008
 - Resharper 4.5
- Hudson
 - Java JRE 1.6

4/11/2010

For transparency, here is what I am using for my demos today.

Agenda | Continuous Integration

Concepts of Continuous Integration

Build Server Scripting

Build Integration

Build Metrics

Build Deployment

4/11/2010

Transition

Build Server Scripting

- Inclusive Solution Structure
- Simple Build Script
- Hudson Project Configuration

4/11/2010

Inclusive Structure

- All assembly and utility dependencies are included in project (No GAC)
- Any machine can build the project (no build setup)

Build Server Scripting

Show Me Code

4/11/2010

- Background Info
 - Subversion Structure
 - Development Repository
 - Vendor Repository
 - Binary Repository
 - Project Structure
 - Inclusive Dependencies
 - Lib (SVN:Externals)
 - Batch Files
- Build Script
 - Planning a conversion to Powershell, once I have time or a really urgent need
 - Build Targets
 - Build Variables
 - Initialize, Compile, Move, Test
- Hudson Project Configuration
 - Subversion Monitoring
 - Running the Build Script
 - Caution on using too many build options – how much do you want to depend on Hudson?
 - Viewing
 - Build Console
 - Project Workspace
 - Change History
- Testing the Build, Commit a Change

Agenda | Continuous Integration

Concepts of Continuous Integration

Build Server Scripting

Build Integration

Build Metrics

Build Deployment

4/11/2010

Transition

Build Integration

- Distributing Dependent Projects
- Linking Hudson Projects

4/11/2010

Build Integration

Show Me Code

4/11/2010

- Project Dependency (Common)
 - Integration Tests (Database Configuration)
 - Distribute (Gold Repository)
- Dependent Project (HelloWorld)
 - Lib (SVN:Externals)
- Hudson Project
 - Build Other Projects (common)
 - Build After Other Projects (HelloWorld) (Automatic Hookup)
- Testing the Build, Commit a Change

Agenda | Continuous Integration

Concepts of Continuous Integration

Build Server Scripting

Build Integration

Build Metrics

Build Deployment

4/11/2010

Build Metrics

- Assembly Versioning
- Code Conventions
- Code Complexity
- Hudson Reports
- Build Tagging

4/11/2010

Build Metrics

Show Me Code

4/11/2010

- Project Build Script
 - Assembly Versioning
 - Namespace Consistency
 - File Headers
 - Static Analysis (FxCop)
 - Code Coverage
- Hudson Reports
 - Unit Tests Results
 - Code Task Scanning
 - Revision Tagging
- Testing the Build, Commit a Change

Agenda | Continuous Integration

Concepts of Continuous Integration

Build Server Scripting

Build Integration

Build Metrics

Build Deployment

4/11/2010

Build Deployment

- Assembly Packaging
- Deployment Scripting
- Scheduling Jobs

4/11/2010

Build Deployment

Show Me Code

4/11/2010

- Project Packaging
 - Packaging Target in Build
- Project Deployment Script
 - For Web Application (Can be done with Windows Services as well)
 - Targets
 - Batch Files
 - Setting up the target Environment
- Scheduled Hudson Projects
- Testing the Deployment, Run the Project

Summary | Continuous Integration

Concepts of Continuous Integration

Build Server Scripting

Build Integration

Build Metrics

Build Deployment

4/11/2010

Resources

Duvall, Paul M. (2007). *Continuous Integration. Improving Software Quality and Reducing Risk*. Addison-Wesley. ISBN 0-321-33638-0.

Hudson Wiki. <http://wiki.hudson-ci.org/display/HUDSON/Meet+Hudson>

Fowler, Martin (2006). *Continuous Integration*. <http://martinfowler.com/articles/continuousIntegration.html>



4/11/2010

Follow-Up

E-Mail todd.gardner@toro.com

Blog <http://ejournal.wordpress.com/>

Twitter toddhgardner

Remember to complete your session evaluation

4/11/2010